THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 61, No. 8, October 1982
Printed in U.S.A.

Spatial Subsampling in Motion-Compensated Television Coders

By J. D. ROBBINS and A. N. NETRAVALI

(Manuscript received December 9, 1981)

Motion-compensated television coders generate data at a nonuniform rate, which is smoothed by a buffer for transmission over a channel of constant bit rate. Spatial subsampling is one of the methods used to prevent overflow of the buffer, which would otherwise occur for scenes with complex motion from frame to frame. In this paper we evaluate the effects of spatial subsampling on the performance of motion-compensated coders. In particular, we find that, although the quality of motion estimation does degrade in the presence of subsampling, the degradation is not substantial. Use of 2:1 horizontal subsampling, for example, results in bit rates that are 50 percent lower compared with no subsampling for motion-compensated coders. This percentage is approximately the same for the conditional-replenishment coders. Spatial subsampling generally results in blurring of the picture. We describe a technique for adaptive interpolation that results in blurring of only the unpredictable areas of the picture. The subjective quality in the presence of subsampling is thus improved considerably. In conclusion, our techniques for subsampling in a motion-compensated coder reduce the bit rate approximately by the same factor as subsampling in a conditional-replenishment coder, but result in a much better picture quality.

I. INTRODUCTION

Television signals contain a significant amount of frame-to-frame redundancy. Interframe coders attempt to exploit this redundancy by

(*i*) Segmenting each television frame into two parts, one part that is predictable from the previous data, and one part that is unpredictable.

(*ii*) Transmitting two types of information: (a) addresses specifying the location of the picture elements in the unpredictable area, and (b)

information (usually quantized prediction error) by which the intensities of the unpredictable area can be updated.

(*iii*) Matching the coder bit rate to the channel rate. Since the motion in a real television scene occurs randomly and in bursts, the amount of information about the unpredictable area will change as a function of time. It is transmitted over a constant bit-rate channel by, (a) storing it in a buffer prior to transmission to smooth out the transmitted information rate, and (b) using the buffer fullness to regulate the encoded bit rate by varying the amplitude, spatial, and temporal resolution of the television signal. Intensities of the picture elements (pels) in the unpredictable areas are transmitted by predictive coding. In conditional-replenishment coding,[1–4] quantized values of frame difference, element difference, and line difference (or a combination thereof) are transmitted. In motion-compensated coders,[5–9] estimates of interframe translation of objects are obtained, and more efficient predictive coding is performed by taking differences of elements from the previous frame that are appropriately translated. The translation is equal to the displacement of the object. In our previous papers[5–9] we described several methods of displacement estimation and locally adaptive prediction to reduce the bit rate of interframe coders. Displacement estimation methods were recursive, which minimized the motion-compensated prediction error by a steepest-descent algorithm.

As mentioned before, most interframe coders require a buffer to smooth the output of the coder for transmission over a channel. One method of reducing the size of the buffer or preventing buffer overflow is to control the resolution by spatial subsampling. Typically, if the buffer starts to fill rapidly, resolution is decreased; resolution is increased if the buffer is nearly empty. It is not known how to optimally control the resolution of the unpredictable area for a given channel-bit rate and buffer size. However, many excellent resolution-channel algorithms have been designed on a trial and error basis. In this paper, we are concerned with spatial subsampling in motion-compensated coders, for the purpose of

(*i*) Investigating to what extent spatial subsampling adversely affects the recursive-displacement estimation algorithm

(*ii*) Modifying the displacement estimator to increase its efficiency in the presence of subsampling

(*iii*) Evaluating a new algorithm for adaptive interpolation that blurs only the unpredictable area (as compared with the "moving area") in a conditional-replenishment coder

(*iv*) Presenting simulation results on synthetic scenes with known displacement, and real scenes containing complex motion.

Our simulations are restricted to horizontal subsampling by factors

of 2:1 and 4:1. Simulations indicate that spatial subsampling does degrade our displacement estimation. However, the degradation is not very serious. It is known (and confirmed by our simulations) that 2:1 and 4:1 subsampling reduces the bit rates of conditional-replenishment coders approximately by a factor of two and four, respectively. We found that in the case of motion-compensated coders, 2:1 and 4:1 subsampling reduces the bit rates of a motion-compensated coder by similar factors. In most conditional-replenishment coders, subsampling blurs the "moving areas" (i.e., pels for which amplitude of the frame difference is above a certain threshold); our adaptive interpolation algorithm blurs only the unpredictable areas. Thus, improvement in prediction reduces the blurred areas, thereby improving the picture quality.

II. ALGORITHM

In this section, we describe the modifications to our displacement estimation algorithm. It is worthwhile, however, to look at the basic algorithm described in our earlier works.[5] Let $I(\mathbf{x}_k, t)$ denote the intensity of a scene at the kth sample point $\mathbf{x}_k$ in the scanning order of a frame, and let $I(\mathbf{x}_k, t - \tau)$ denote the intensity at the same spatial location in the previous frame. If the scene consists of an object that is undergoing pure translation under uniform illumination, then, disregarding the background,

$$I(\mathbf{x}_k, t) = I(\mathbf{x}_k - D, t - \tau), \tag{1}$$

where $\mathbf{D}$ is the displacement (two-component vector) of the object in one frame interval, τ. The pel-recursive algorithm obtains an estimate of $\mathbf{D}$ (i.e., $\hat{\mathbf{D}}$) by recursively minimizing the square of the displaced-frame difference at the current pel location. The displaced-frame difference $DFD(\cdot, \cdot)$ is defined by

$$DFD(\mathbf{x}_k, \hat{\mathbf{D}}) = I(\mathbf{x}_k, t) - I(\mathbf{x}_k - \hat{\mathbf{D}}, t - \tau)\,. \tag{2}$$

The minimization is performed by a steepest-descent algorithm of the form

$$\hat{\mathbf{D}}_{k+1} = \hat{\mathbf{D}}_k - \tfrac{1}{2}\epsilon\nabla_{\mathbf{D}}[DFD(\mathbf{x}_{k+1}, \hat{\mathbf{D}}_k)]^2\,, \tag{3}$$

where $\nabla_{\mathbf{D}}[\cdot]$ is the two-dimensional gradient with respect to $\mathbf{D}$. Equation (3) can be expanded to

$$\hat{\mathbf{D}}_{k+1} = \hat{\mathbf{D}}_k - \epsilon DFD(\mathbf{x}_{k+1}, \hat{\mathbf{D}}_k)\nabla I(\mathbf{x}_{k+1} - \hat{\mathbf{D}}_k, t - \tau)\,, \tag{4}$$

where $\nabla = \nabla_x$ is the two-dimensional spatial-gradient operator with respect to horizontal and vertical coordinates of vector $\mathbf{x}$. We use a finite-difference approximation for the gradient, which is formed by

element difference, $EDIF_k$, and line difference, $LDIF_k$, using the pel closest to the point $\mathbf{x}_{k+1} - \mathbf{D}_k$ in the previous frame. The above displacement estimator requires multiplication at each iteration, which is undesirable for hardware implementation and is therefore simplified to:

$$\hat{\mathbf{D}}_{k+1} = \hat{\mathbf{D}}_k - \epsilon \operatorname{sgn}|DFD(\mathbf{x}_{k+1}, \hat{\mathbf{D}}_k)| \cdot \operatorname{sgn}|\nabla I(\mathbf{x}_{k+1} - \hat{\mathbf{D}}_k, t - \tau)|, \quad (5)$$

where

$$\operatorname{sgn}(z) = \begin{cases} -1, \text{ if } Z < -T \\ 0, \text{ if } |Z| \leq T \\ +1, \text{ otherwise.} \end{cases} \quad (6)$$

The above recursion to update $\hat{\mathbf{D}}_k$ is carried out only in the moving areas of the current frame, i.e., for those pels where

$$\sum_{j=-p}^{+p} |I(\mathbf{x}_{k+j}, t) - I(\mathbf{x}_{k+j}, t - \tau)| \geq \text{Threshold}.$$

The motion-compensated coder predicts intensity, $I(x_k, t)$, using either the previous frame intensity, $I(\mathbf{x}_k, t - \tau)$, or displaced-previous-frame intensity, $I(\mathbf{x}_k - \hat{\mathbf{D}}_{k-1}, t - \tau)$, based on which predictor results in less error for certain already transmitted neighbors. We note that if a point $\mathbf{x} - \mathbf{D}$ does not lie on the grid formed by the pels, then interpolation is required to evaluate $I(\mathbf{x}_k - \hat{\mathbf{D}}_{k-1}, t - \tau)$. The displacement at either the previous pel or the previous line element is used to form the displaced-previous-frame prediction of the present pel. This allows the receiver to compute displaced-previous-frame predictions without explicit transmission of the displacement. If the magnitude of the prediction error exceeds a predetermined threshold, the coder transmits a quantized version of the prediction error and the necessary addressing information to the receiver.

Above we described the basic motion-compensation algorithm. We now describe its modifications relevant to the spatially subsampled television signal. Although the algorithm we describe below can be applied to signals subsampled in a variety of ways, we restrict ourselves to horizontal (along a scan line) subsampling by a factor of two and four to one. Modifications and details of each component of the motion-compensated system are given below.

2.1 Displacement estimator

Figure 1 shows the pel arrangement used for the updating process of the displacement estimator. Since the subsampled and subsequently interpolated pels contain more noise, they are given less importance in the updating process. This is done in two ways. Tbe frame-difference signal at subsampled pels is given less weight in determining where

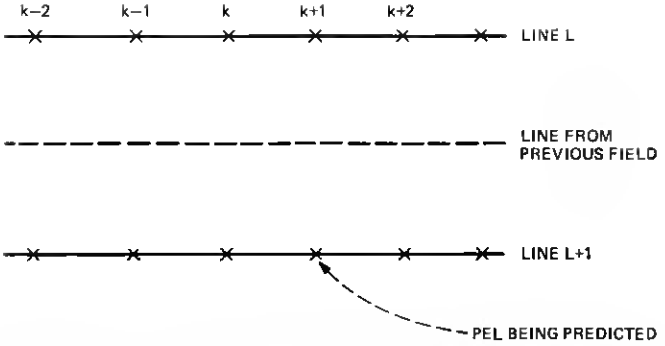

Fig. 1—Pel configuration for a displacement-updating process. Displacement is recursively updated only at those pels where the weighted sum of frame difference in a window around pel k is above a threshold.

the displacement is updated and a smaller updating constant ('epsilon') is used at subsampled pels. Thus, the estimator works as follows:

$$\hat{\mathbf{D}}_k = \hat{\mathbf{D}}_{k-1} - \epsilon SGN|\, DFD(\mathbf{x}_k, \hat{\mathbf{D}}_{k-1})\,| \cdot SGN \left| \begin{matrix} EDIF_k \\ LDIF_k \end{matrix} \right|, \tag{7}$$

where

$$\epsilon = \begin{cases} \epsilon_1, \text{ if pel } x_k \text{ is subsampled} \\ \epsilon_2, \text{ otherwise .} \end{cases} \tag{8}$$

We have found that the estimator performance is improved by choosing $\epsilon_1 < \epsilon_2$ (both positive numbers). The recursion is carried out only at those pels where a weighted sum of the magnitude of the frame difference in a window around pel k exceeds a given threshold. Thus, $\hat{\mathbf{D}}_{k-1}$ is updated only if

$$\sum_{j=-p}^{p} w_j |\, FDIF(\mathbf{x}_{k+j})\,| \geq THRESH_1\,; \tag{9}$$

otherwise $\hat{\mathbf{D}}_k = \hat{\mathbf{D}}_{k-1}$. The weights $\{w_j\}$ are nonnegative and are lower for subsampled pels compared with the nonsubsampled pels. The threshold, $THRESH_1$, is preselected and optimized by simulations, and the displacement $\hat{\mathbf{D}}_k$ was used for the prediction of the element at location k in the next line (see Fig. 1).

2.2 *Predictor and predictor selection*

As in our earlier works, we have used only predictors based on intensities in the previous frame. The previous-frame and displaced-previous-frame predictions are calculated for each pel, whether subsampled or not. Of course, other predictions (e.g., intrafield predictors such as previous element or line) can be used to augment our prediction strategy. Having computed both the previous-frame and displaced-

frame predictors, we use the following rule to switch adaptively between them on a pel-by-pel basis. Referring to Fig. 1, we use the frame-difference predictor if

$$\sum_{j=-m}^{+m} w_j |FDIF(\mathbf{x}_{k+j})| \leq \sum_{j=-m}^{+m} w_j |DFD(\mathbf{x}_{k+j}, \hat{\mathbf{D}}_k)|; \tag{10}$$

otherwise we use the displaced-frame predictor. The above inequality is evaluated for a window of size $(2m + 1)$ pels centered around pel k. Again, less weight is given to pels that are subsampled, i.e., w_j is lower for subsampled pels. In the calculation of $FDIF(\cdot)$ and $DFD(\cdot, \cdot)$, sometimes the use of interpolated pels in the previous frame may be required.

2.3 Subsampling and interpolation

We considered several patterns for subsampling. Some patterns were such that they did not change from line to line, field to field, or frame to frame, whereas some were intentionally staggered. Some of these are described in Section 2.4 Having selected a subsampling pattern, we then interpolated the missing pels using an adaptive technique. If the magnitude of the prediction error for the nearest nonsubsampled pels to the right and left was below a threshold, then the intensity of the subsampled pel was replaced by its prediction. If, on the other hand, either the closest right or left nonsubsampled neighbor had prediction-error magnitude above the threshold, then a simple linear (horizontal) interpolation was used to reconstruct the intensity of the subsampled pel. The threshold used is the same as the one that determines whether the quantization error is transmitted. This type of adaptive interpolation improves with the quality of prediction. The conditional-replenishment coders subsample the "moving area" pels, which are determined by the frame difference signal. This has the effect of blurring the entire moving area even if more sophisticated predictors are used. Our strategy, on the other hand, replaces all the "predictable" pels (as determined by the closest neighbors) by their prediction rather than by spatial interpolation. Thus, only the unpredictable areas are blurred by spatial interpolation.

2.4 Transmitted information

As previously mentioned, we transmitted to the receiver the quantized prediction error of every nonsubsampled pel where magnitude of prediction error was above a threshold, called the replenishment threshold. This classifies pels into predictable and unpredictable pels. A 35-level symmetric quantizer was used with representative levels at 0, 3, 6, 11, 16, 21, 28, 35, 44, 53, 64, 77, 92, 109, 128, 149, 178, and 197

(on an 8-bit scale of 0 to 255). We obtained decision levels by averaging the adjacent representative levels. A code set for representing the quantizer levels was not designed, but entropies were computed. In addition, horizontal run lengths of predictable and unpredictable pels were transmitted. Here again, entropies of the predictable- and unpredictable-pel run lengths were calculated. This assumes that in practice separate code sets will be used for run lengths of predictable and unpredictable pels.

III. SIMULATIONS AND RESULTS

Computer simulations were performed on two types of scenes. The first was a synthetic scene that was computer-generated. It was a damped radial cosine in intensity with a radius of 60 pels, which translated from frame to frame by a given amount. The pattern is described mathematically by the intensity function

$$I(R) = 100 \cdot \exp(-0.01R)\cos(2\pi R/P),\ 0 \leq 60\,,$$

where R is the radial distance from the center [taken to be (100,100)] and

$$P = (1 - R/60)10 + 10\,.$$

This function is displayed on a 256- by 256-element raster in two interlaced fields of 128 lines each. The pattern is shown as Fig. 6 in Ref. 7. The other scene, called Judy, is a head and shoulders view of a person engaged in active conversation. This consisted of 50 frames obtained by taking a Nyquist-rate sampling of a video signal having a 1-MHz bandwidth. Each sample was quantized uniformly to 8 bits. Four frames of this sequence are shown in Fig. 4 of Ref. 5.

3.1 Synthetic Scene

The simulations on the synthetic scene were restricted to 2:1 subsampling with a subsampling pattern that did not change from line to line and field to field (referred to as a nonstaggered pattern). The purpose of this simulation was to evaluate the degradation in the performance of the displacement estimator. Therefore, only the displacement was calculated, without using it for coding. Figures 2 and 3 show the relative displacement error as a function of the iteration number. The iteration number in this case refers to only those instances where the displacement was actually updated. Owing to several factors [e.g., the setting of the threshold in eq. (9)], a given iteration number in a subsampled case may not be at the same location in the nonsampled case. Figure 2 shows the case when the pattern moves at 4 pels per frame, whereas Fig. 3 shows the results for displacement of 5 pels per frame. The parameters of the displacement estimator (of

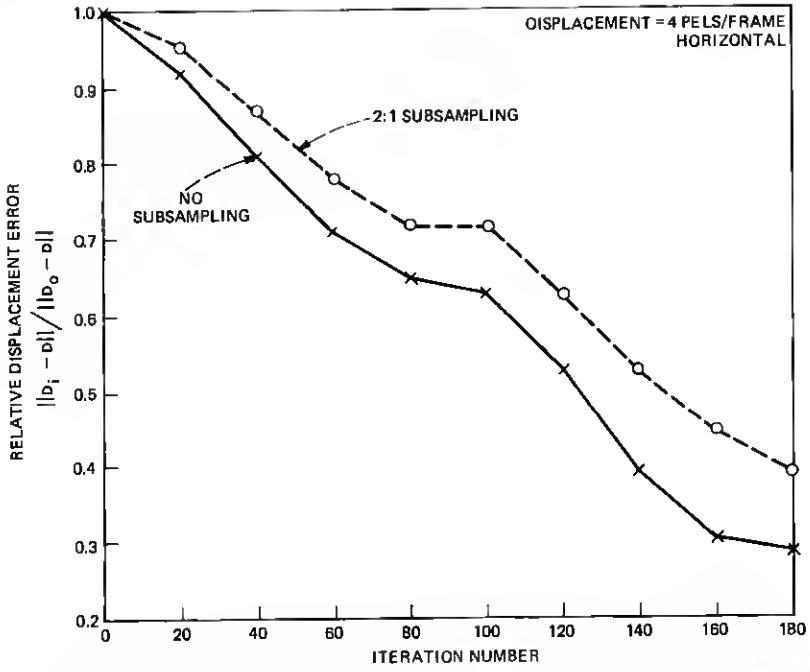

Fig. 2—Relative error in displacement for a synthetic moving pattern at 4 pels per frame as a function of iteration number.

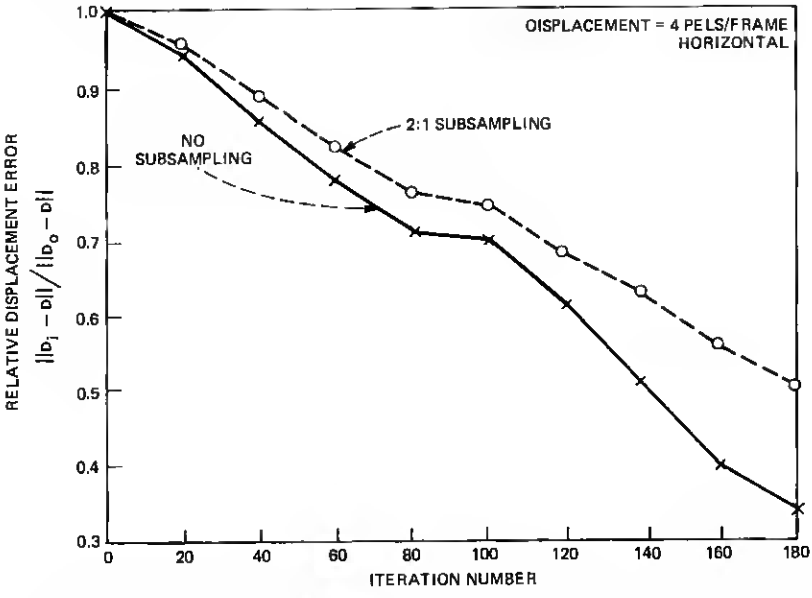

Fig. 3—Relative error in displacement for a synthetic moving pattern at 5 pels per frame as a function of iteration number.

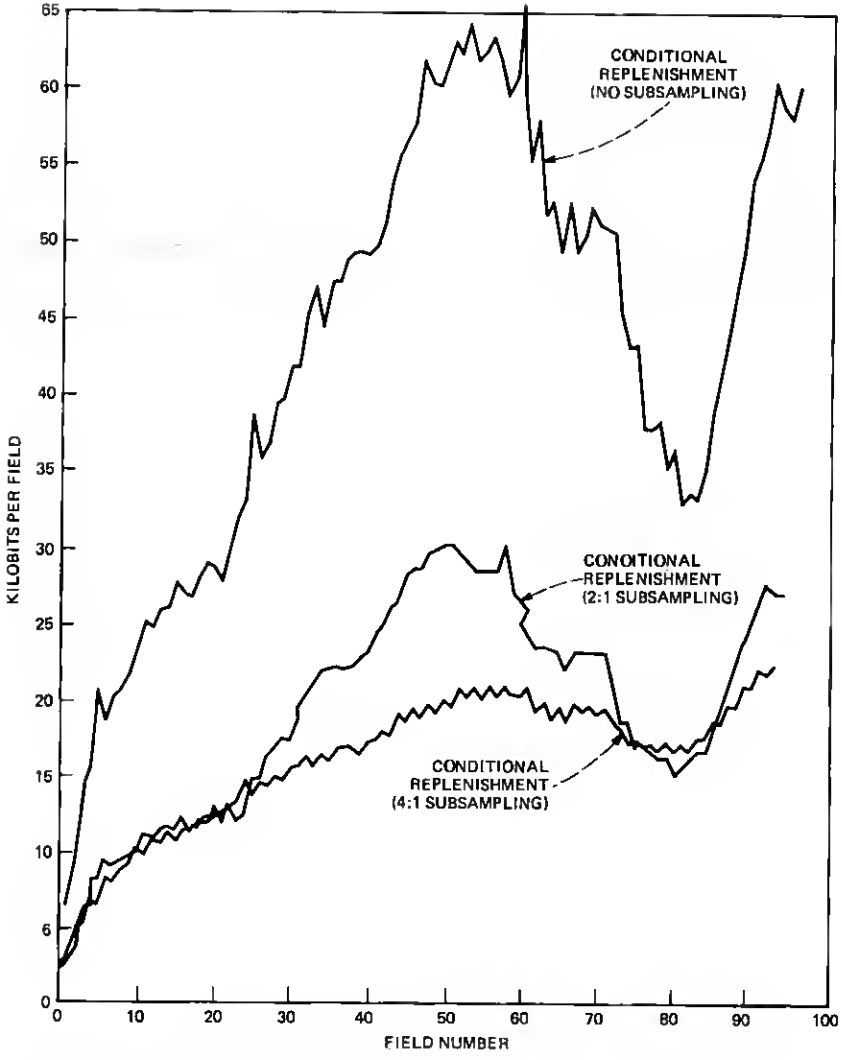

Fig. 4—Bits required per field for conditional-replenishment scheme using a fine (511-level) quantizer.

Section II) were selected by trial and error. It is seen from both figures that the convergence of the displacement estimator is more rapid when there is no subsampling. The degradation appears to be somewhat less for the 5-pel/frame case as compared with the 4-pel/frame case. The effect of this degradation on the motion-compensated coder is evaluated in Section 3.2.

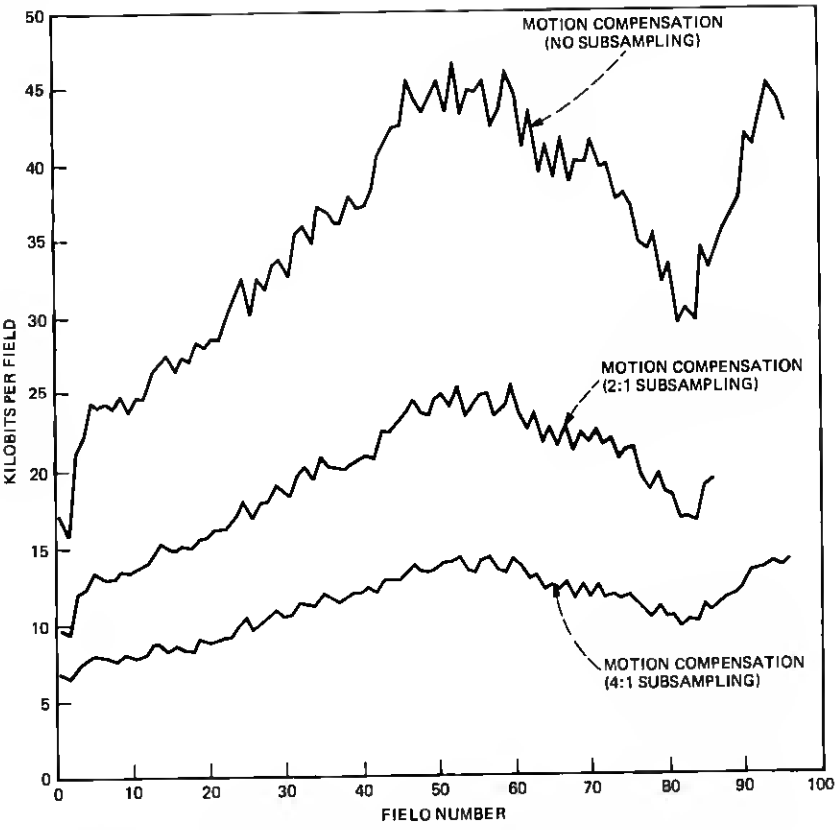

Fig. 5—Bits required per field for motion-compensation scheme using a fine (511-level) quantizer.

3.2 Real Scene

Before the performance of the subsampled motion-compensated coders is given, it is important to note the parameters that were chosen for simulations. These choices were based on trial and error. While this may not have resulted in optimum choices, our choices may not be too far from the optimum.

The epsilon of eq. (8) was different for subsampled pels compared with nonsubsampled pels; ϵ_1 was $\frac{1}{64}$, and ϵ_2 was $\frac{1}{32}$. The parameters in the update condition of eq. (9) were taken to be: $p = 1$, and $THRESH_1 = 4$ (on a scale of 0 to 255, 8-bits).The weight given to subsampled pels was 1 and to nonsubsampled pels was 2. Thus, through ϵ and these weights, subsampled pels were given less "importance" in the displacement estimation process. The predictor selection [eq. (10)] was done by using a window of 3 (i.e., $m = 1$), and the subsampled pels were given weight $w_j = 1$, whereas nonsubsampled pels were given weight

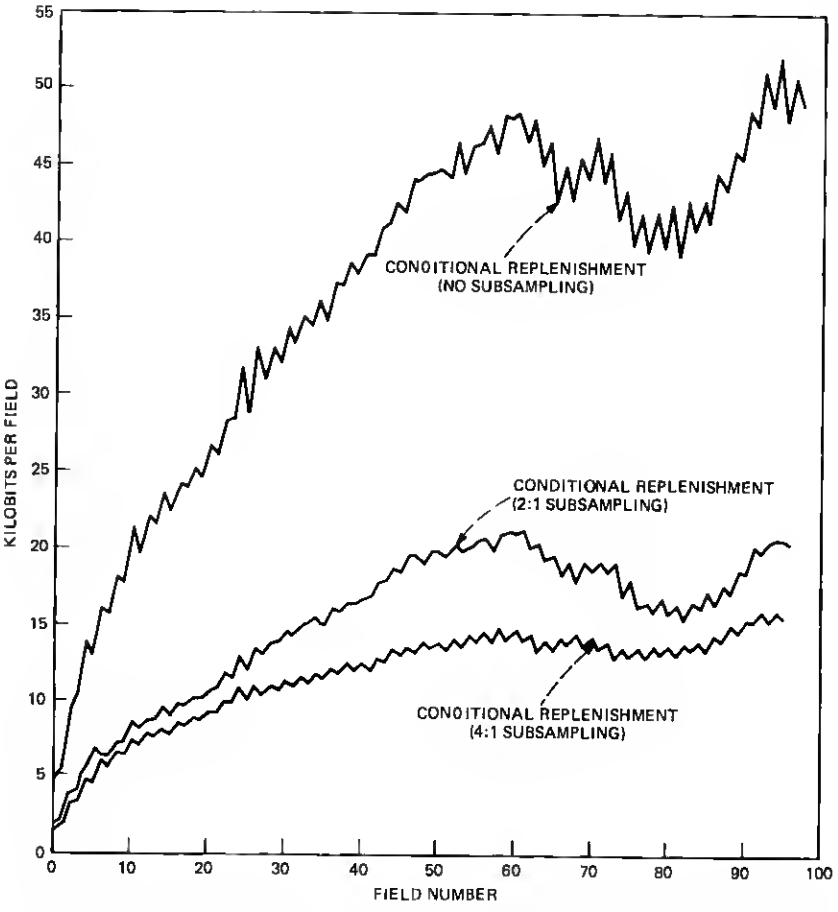

Fig. 6—Bits required per field for conditional-replenishment scheme using a coarse 35-level quantizer.

$w_j = 4$. The displacement estimator was not initialized at the beginning of each scanning line; thus, the estimate from the last pel of the previous line was used as the initial estimate for the first pel of the next line.

The performance was measured by the number of coded bits that were required for each field (approximated by appropriate entropies). When the coarse quantizer of Section 2.4 was used, then the total squared-interpolation error was also calculated for each field. Figures 4 and 5 show the coded bits for both conditional replenishment and motion compensation, when no coarse quantization was performed (i.e., quantizer with 511 levels was used). It is obvious that 2:1 subsampling reduces the bits/field by approximately two for both conditional

replenishment and motion compensation. The decrease in bit rates is high for those fields with a large amount of motion. The 4:1 subsampling reduces the bit rates of motion-compensation schemes much more significantly compared with conditional replenishment. However, this may be a peculiarity of the particular scene we used for simulation. We used a few more scenes and found that 4:1 subsampling, in general, reduced the rate by a factor of two compared with 2:1 subsampling for both conditional replenishment and motion compensation.

Using the 35-level quantizer mentioned earlier, the bit rates are plotted in Figs. 6 and 7. Figure 6 shows conditional replenishment and Fig. 7 shows motion compensation. These figures indicate that motion compensation results in approximately 60-percent reduction for no subsampling, approximately 80-percent reduction for 2:1 subsampling, and about 90-percent reduction for 4:1 subsampling, compared with

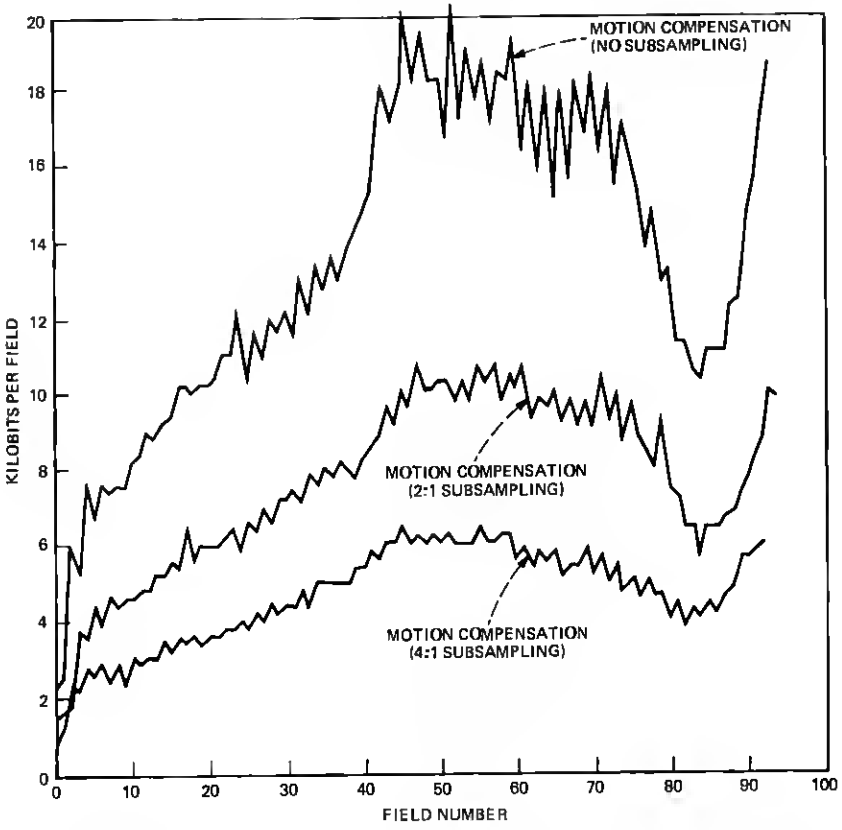

Fig. 7—Bits required per field for motion-compensation scheme using a coarse 35-level quantizer.

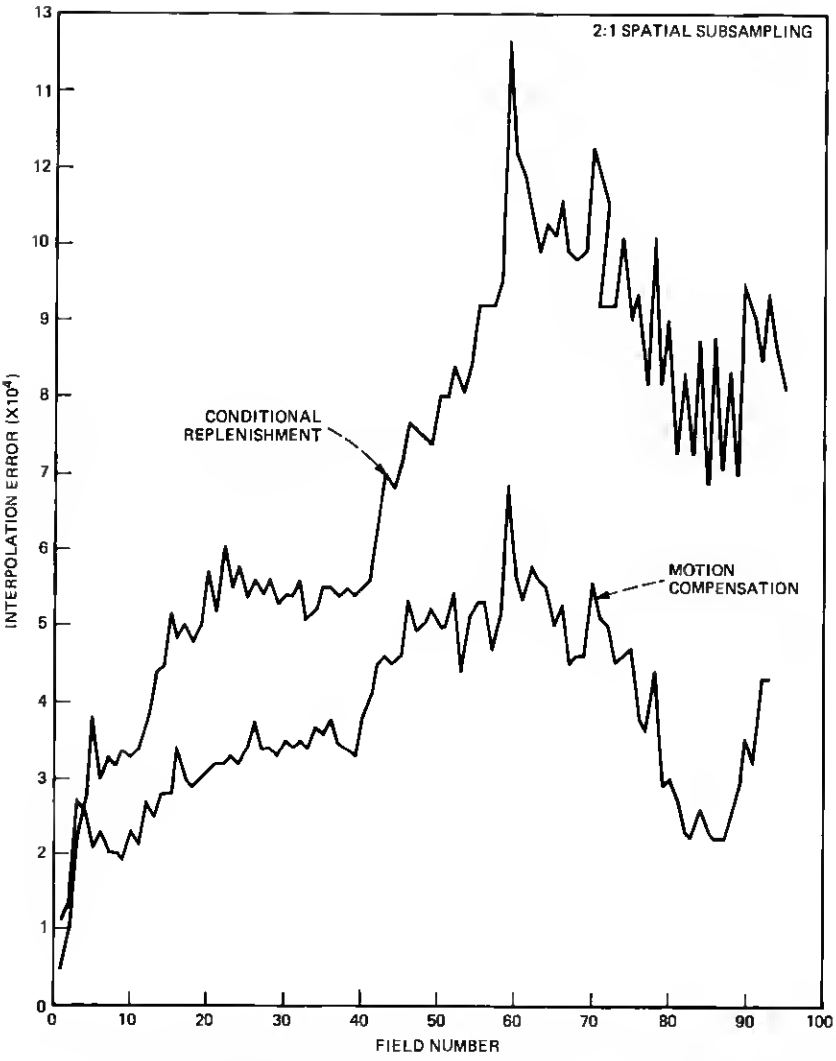

Fig. 8—Plot of squared-interpolation error summed over entire field versus field number of 2:1 spatial subsampling. Adaptive interpolation is used in the case of motion compensation. Fixed-linear one-dimensional spatial interpolation is used for conditional replenishment.

conditional replenishment. Obviously these conclusions are scene-dependent. However, for scenes containing significant translational motion, such conclusions may remain valid. It is always difficult to evaluate quality of short segments of scenes. We made informal observations to compare pictures resulting from conditional replenishment and motion compensation. Subsampling results in visible blur-

ring. However, it was found that, owing to our adaptive interpolation scheme, motion compensation blurred a much smaller area than did conditional replenishment. Also, the blurred areas in motion compensation appear to be much more fragmented and somewhat randomly distributed, which also decreases their visibility. Figures 8 and 9 show the plots of squared-interpolation error per field versus the field number for both 2:1 and 4:1 subsampling. Curves for both frame-difference conditional replenishment and motion compensation are shown. In the case of 2:1 subsampling, the interpolation error decreases by almost a factor of two using motion compensation. This decrease is even greater for 4:1 subsampling.

In our simulations we also tried staggering the subsampling pattern

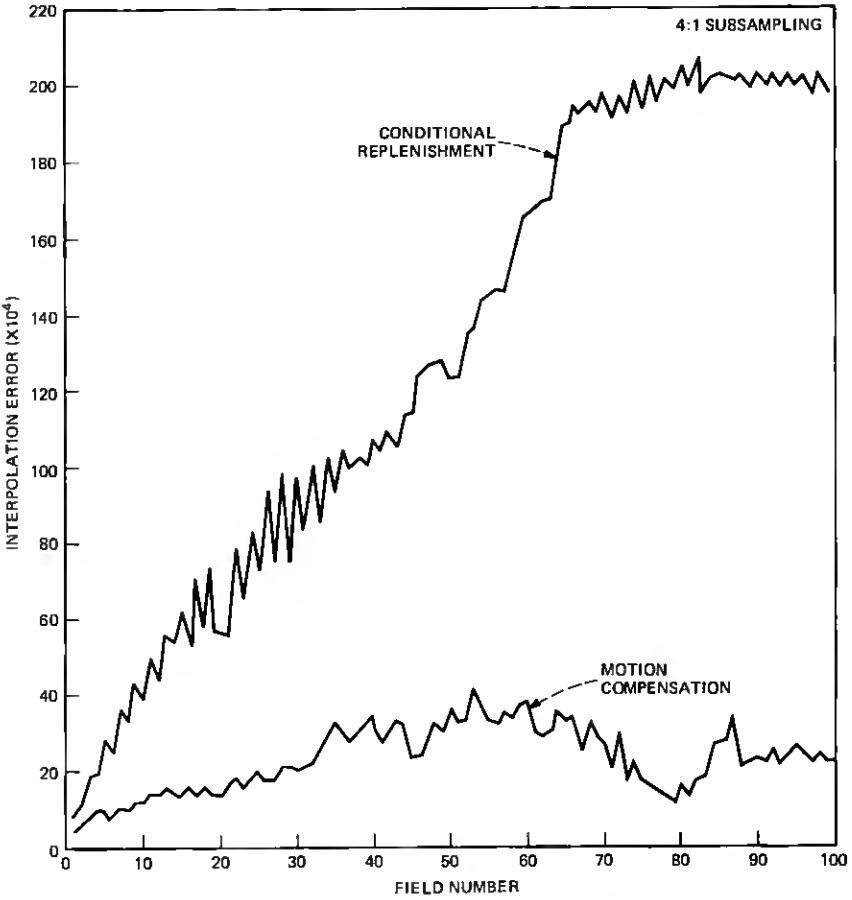

Fig. 9—Plot of squared interpolation error summed over an entire field versus field number for a 4:1 spatial subsampling.

from line to line, field to field, and frame to frame. This makes the quantization and the interpolation noise appear random and less patterned. However, field-to-field or frame-to-frame staggering results in annoying flicker, as expected. We thus found line-to-line staggering to be most useful. Although staggering improved the quality of pictures for both conditional replenishment and motion compensation, the improvement was somewhat higher for conditional replenishment. The required bits per field did not change significantly because of staggering. Thus, we might conclude that line-to-line staggered-subsampling patterns improve the quality of pictures without any significant increase in the bit rates.

IV. SUMMARY AND CONCLUSIONS

We have presented in this paper schemes for motion compensation in the presence of spatial subsampling, which is required in interframe coders to prevent buffer overflow.We also described an adaptive interpolation scheme that blurred only the "unpredictable" area during subsampling. Computer simulations were performed in synthetic scenes to evaluate degradation of the displacement estimator in the presence of subsampling. It was found that, although the quality of displacement estimation was degraded in the presence of spatial subsampling, the effect on the bit rates was not significant. Compared with conditional replenishment, motion compensation reduced the bit rates by a factor of two or more even during subsampling. Thus, a 2:1 subsampled motion-compensated coder results in about one quarter of the bit rate of conditional replenishment and about one half of the bit rate of the 2:1 subsampled conditional-replenishment coder. Our adaptive interpolation scheme blurs only the "unpredictable area" rather than the "moving area" that is blurred in subsampled conditional-replenishment coders. Since "unpredictable area" is a subset of moving area and is fragmented randomly, the blurring caused by subsampling in the case of a motion-compensated coder is much less visible. This is also borne out by the total-interpolation error, which decreases by more than a factor of two using adaptive interpolation.

REFERENCES

1. F. W. Mounts, "A Video Encoding System Using Conditional Picture-Element Replenishment," B.S.T.J., *48*, No. 7 (September 1969), pp. 2545–54.
2. B. G. Haskell, F. W. Mounts, and J. C. Candy, "Interframe Television Coding of Videotelephone Pictures," Proc. IEEE, *60*, No. 7 (July 1972), pp. 792–800.
3. T. Ishiguro, K. Iinuma, Y. Iijima, T. Koga, S. Azaini, and T. Mune, "Composite Interframe Coding of NTSC Color Television Signals," 1976 Nat. Telecommun. Conf. Rec., Vol. *1*, Dallas, Texas, November 1976, pp. 6.4-1 to 6.4-5.
4. B. G. Haskell, "Frame Replenishment Coding of Television," in *Image Transmission Techniques*, W. K. Pratt, ed., New York: Academic Press, 1978.
5. A. N. Netravali and J. D. Robbins, "Motion-Compensated Television Coding: Part 1," B.S.T.J., *58*, No. 3 (March 1979), pp. 631–70.

6. J. D. Robbins and A. N. Netravali, "Interframe Coding Using Movement Compensation," Int. Commun. Conf. (June 1979), pp. 23.4/1-5.
7. J. A. Stuller and A. N. Netravali, "Transform Domain Motion Estimation," B.S.T.J., *58*, No. 7 (September 1979), pp. 1673–703.
8. A. N. Netravali and J. A. Stuller, "Motion-Compensated Transform Coding," B.S.T.J., *58*, No. 7 (September 1979), pp. 1703–18.
9. A. N. Netravali and J. D. Robbins, "Motion-Compensated Coding: Some New Results," B.S.T.J., *59*, No. 4 (November 1980), pp. 1735–45.